Autonomous Parallel Parking Methodology for Ackerman Configured Vehicles.

Ankit Gupta[1, 2], Rohan Divekar[1]
[1]Department of Electronics and Telecommunication, Maharashtra Institute of Technology,
Paud Road, Kothrud, Pune, Maharashtra, India – 411038
ankit_gupta@ieee.org
divekarrohan@gmail.com
[2]Advance Robotics Research Organization
ankit@arro.in

***Abstract* – Parallel parking is challenge for all drivers say amateurs or the experts. An automatic car parking system is a solution to this ordeal. This vehicular technology has been implemented using many other systems but a cost effective, simple and accurate solution will be greatly appreciated. This paper explains in detail a simple and precise autonomous car-parking algorithm for Ackerman steering configuration. A two-part trajectory-planning algorithm consists of the steering planning and simple distance calculations. The limits of vehicle mechanism and drive torque are taken into account. Simulation results are presented to illustrate the application of the proposed algorithm. The algorithm uses simple geometry for its path planning and odometry. The system uses sonar sensors and wheel encoders for its perception. This sensed data is interpreted in the processor of the vehicle. As per the trajectory determined, the vehicle parks itself into the parking space. Simulation and experiment results shows that using this algorithm the parking maneuver will become more safe, efficient and fast.**

***Keywords* – Automatic Parking System, Path Planning, Tracking Algorithm, Parallel Parking, Parking Sensors.**

I.INTRODUCTION

Now a day's parking space has become too scarce in big cities. For amateur drivers to squeeze their cars in such a tiny place is a big nuisance. This often leads to minor dents and scratches on the car. Therefore automatic parking is one of the growing technologies that aim at enhancing the comfort and safety of driving. This system helps drivers to automatically maneuver their vehicles in constrained parking environments where much attention and experience is required. Further it also ensures efficient management of the parking space and time by avoiding traffic congestion. The parking tactic is accomplished by the control of the steering angle and taking into account the original environment conditions for collision-free motion with in the space. Numerous efforts by various automobile researchers and manufacturers are made in this area. Many of these systems involve imaging and complex processing [1]. A commercial version of automatic parallel parking was introduced by Toyota Motor Corporation in Toyota Prius in 2004. Lexus also debuted a car, the 2007 LS, with an Advanced Parking Guidance System.

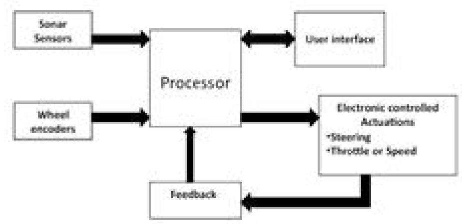

Fig. 1 Block Diagram of Automatic Parking System (APS)

This paper presents a technique that has easy and simple yet effective path planning and tracking control algorithm to automatically park a vehicle. Our approach consists of user interface, ultrasonic sensor data, and wheel encoder data, drive by wire and path planning and path tracking for parallel parking. Fig. 1 shows the block diagram of the system.

II. PATH PLANNING AND KINEMATICS

The path planning involves simple geometrical equations in this system. The path that the vehicle travels before maneuvering into the parallel parking place, perfectly aligned, has three differentiable segments to consider. One is the straight line and the other two are the arcs of circles, as shown in fig. 2. During the whole parking task the wheel has to align and change its angle only twice, at point 'p' and point 'o'. This not only shunts the possible errors that could arise from frequent steering but also consumes less power and hence is more energy efficient and simple. The whole parking trajectory is calculated only from the knowledge of the distance between the parking vehicle and the vehicle already parked which is obtained from the distance sensors. All other parameters required for path planning are either constant or are derived from the above-mentioned distance parameter using equations, explained further in this section. This illustrates the fact that not much sensing is required for the path planning and hence the processing is much simpler.

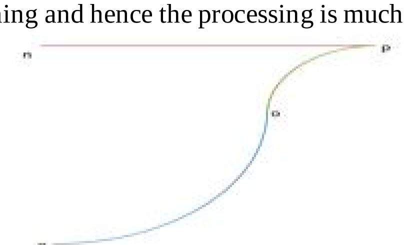

Fig. 2 Parking Trajectory

DOI: 01.IJCSI.02.02.130

This trajectory for parking does not demands the vehicles to be parallel instead it also perfectly works fine if the vehicles have some angle of misalignment between them. It has some boundary conditions and is very robust.

A. *Ackerman steering calculations.*

Ackerman steering was developed around 1800 A.D. The concept of Ackerman is to have all four wheels roll without slipping, around a common point during a turn [2]. This common point is known as instantaneous center of curvature (ICC).

$$R_{s1} = (l / \tan \grave{e}_{i1}) + d / 2. \qquad (1.a)$$

$$R_{s2} = (l / \tan \grave{e}_{i2}) + d / 2. \qquad (1.b)$$

Where, R_{s1} is first steer angle radius of curvature; *l* is length between front axel and the rear axel; d is the distance between points of contact of rear wheels; $\grave{e}_{i1}$ is the angle of inner front wheel, see fig. 3.

In the (1), *l* and *d* are constant and by substituting the value of $\grave{e}_{i1}$ (that is assumed to be 45^0 in our case) R_{s1} is calculated. R_{s1} varies according to vehicle size for fixed $\grave{e}_i$.

$$\cot \grave{e}_i - \cot \grave{e}_o = d / l \qquad (2)$$

Equation (2) relates $\grave{e}_i$ to $\grave{e}_o$. Where, $\grave{e}_o$ is angle of alignment of outer wheel.

B. *Parking Path Geometry*

1 Ideal Case, when the car is parallel to the parked car.

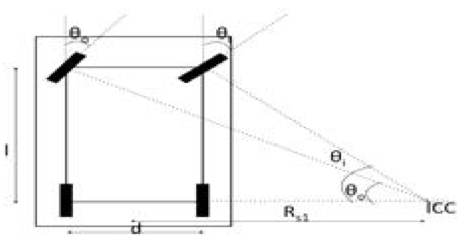

Fig. 3 Ackerman Steering Angle and Geometry.

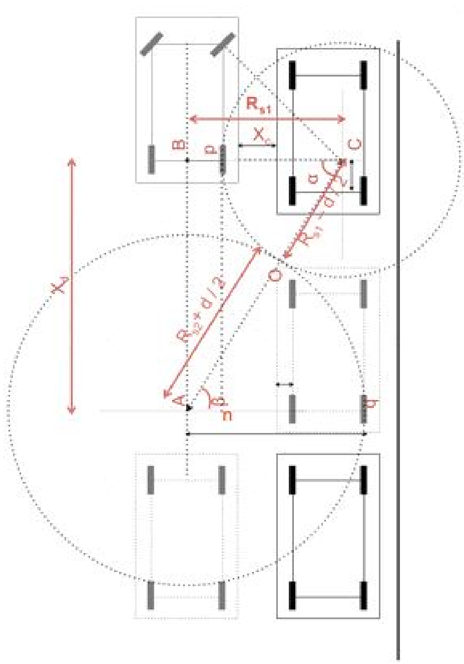

Fig. 4 – Path Planning and Parking Trajectory

When the vehicle is parallel to the parked vehicle the distance between the two vehicles is calculated say X_c using the distance sensors. Then the distance 'A-q' is evaluated using,

$$\text{'A-q'} = d_o + X_c. \qquad (3)$$

Where, d_o is the width of the car. Also,

$$\text{'A-q'} = R_{s2} + d / 2. \qquad (4)$$

Here R_{s2} is the radius of curvature corresponding to second steer angle. Then the vehicles moves to point 'A' which must always line on the line joining the point of contact of rear wheels and extended. This line is speculated from the desired position of the vehicle after complete parking.

Now in right angled triangle 'ABC' of fig. 4,

$$(X_d)^2 = (AB)^2 = (AC)^2 - (CB)^2 \qquad (5)$$

The X_d is the distance that the vehicle needs to drive from point A before turning the steering angle. AC will be $R_{s1} + R_{s2}$. CB is R_{s1}. Now the vehicle moves the distance X_d to point B that is fed back to the system through the wheel encoders this completes the first part of the trajectory indicated with red line in fig. 2. While moving to point B, the system measures the distance between the parked vehicles i.e. parking space. If this distance is less than Aq + FOS (factor of safety) then the system aborts the parking mission and returns back to manual mode.

After reaching point B the wheel is aligned at 45^0.

Angle α is,

$$\alpha = \tan^{-1}(X_d / R_{s1}) \qquad (6)$$

$$\beta = \alpha. \qquad (7)$$

Therefore the arc length 'po' of circle C of radius R_{s1} - d / 2 is,

$$\text{length(po)} = (\alpha / 360) \times 2\pi \times (R_{s1} - d / 2) \qquad (8)$$

arc length 'oq' of circle A of radius R_{s2} + d / 2 is,

$$\text{length(oq)} = (\beta / 360) \times 2\pi \times (R_{s2} + d / 2). \qquad (9)$$

The vehicle then moves the distance 'po', completing the second segment of its trajectory indicated green in fig. 2. At point 'o' the angle of alignment of the wheel is changed to an angle θ_{i2} determined from R_{s2} using (1.a, 1.b) and (2).

After aligning its wheel to the new angle θ_{i2} the vehicle travels arc length 'oq' to conclude its mission with this final segment of its path indicated blue in fig.2.

2. Practical case, when the car is misaligned to the parked vehicle.

Unlike in ideal case, in practical environment it is very likely that during parking the vehicle is not initially aligned parallel to the parked vehicle. In that case the angle of inclination is determined with reference to the parked car using distance sensors as shown in the fig. 5.

$$\ddot{O} = \tan^{-1}((X_{s2} - X_{s1}) / d_s). \qquad (10)$$

Where, d_s is the distance between the front and rear sensor. X_{s2} and X_{s1} are the distances measured by the sensors and Φ is the angle of misalignment.

DOI: 01.IJCSI.02.02.130

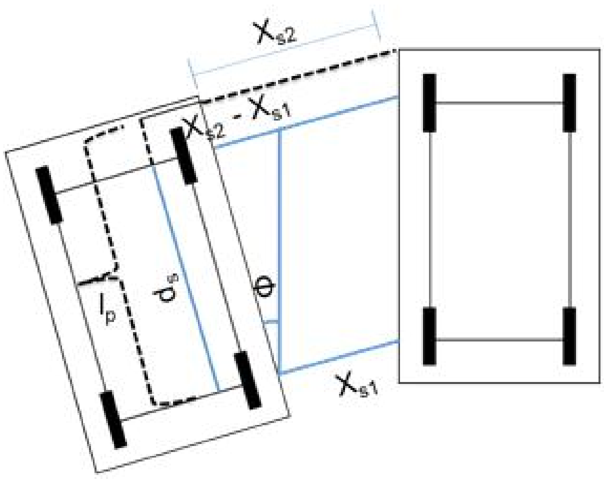

Fig. 5 – Misalignment between the vehicles.

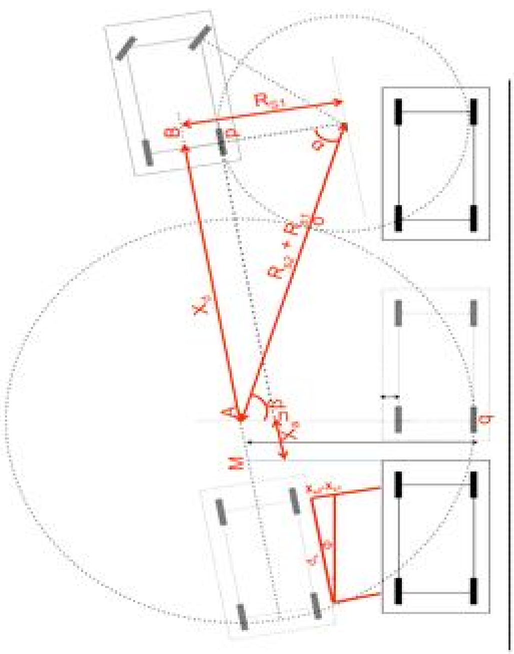

Fig. 6 – Parking Trajectory for a misaligned vehicle.

Once the Ö is calculated form equation (3), then rest of the calculations are analogous to the ideal case, taking into consideration the effect of Ö on other parameters like parking space between the vehicles, the driving distance X_d, localization of the point A (precisely depending on measured values of X_{s1}, the movement l_p and the calculated Ö) and the radius of curvature 'R_{s2}' for third segment of the trajectory. Then the vehicle moves on a three-segment trajectory 'np', 'po' and 'oq' as shown in fig. 6. The calculations are complementary if the angle of misalignment is negative i.e. when vehicle is aligned towards the parked vehicle. The â will be á ± Ö depending on positive or negative misalignment. There are certain boundary conditions on the angle of alignment between the vehicles. The driver is instructed to keep its vehicle with in these conditions in case of violation.

III Electronics and control

The electronics and control system is divided into four subsystems.
Perception devices, the processing unit, actuating devices and the user interface (see fig. 1).

A. *Perception and Sensing components*

These components gather data for the parameters required for the path planning and path tracking. The two sensors

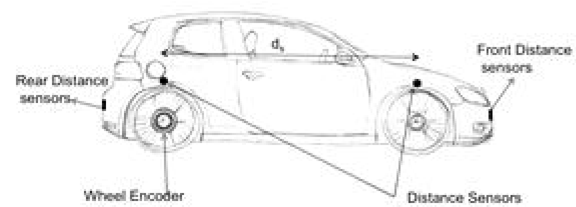

Fig. 7 - Placement of sensors on the vehicle

employed in this system are distance sensor and rotary encoders.The ultrasonic distance sensors are active type sensors. This sensor determines the perpendicular distance of an object or a body from the point of their placement, from the time of flight [3]. These sensors are placed on the outer body above the wheels symmetrically on either side, as illustrated in fig. 7. In the proposed system this sensor is used determine the distance between vehicles 'X_c', the angle of misalignment ±Φ and the parking space availability. Apart from this the distance sensors at rear and front bumpers give the knowledge of the interruptions that might occur during the parking mission, to avoid collisions and ensure a safe maneuver. The high-resolution dual track magnetic wheel encoder is used for path tracking [4]. The encoder provides the information about the rotation of the wheel that is interpreted into distance by a mathematical relation and dead reckoning. This calculations of distance travelled from the rotational data considers the diameter of the wheel for different air pressures for its precise interpretation. These encoders are present in the vehicle for Anti-lock Breaking System (ABS) and Electric Stability Control (ESC). A 640-pole pair encoder gives a resolution of 0.12^0.

B. *The Processing Unit.*

This is the brain of the system that establishes a link between all other subsystems of the system (see fig. 1). From the data available from the sensors the path planning is performed at this block of the system. Then the processor commands the actuating units to maneuver the calculated trajectory and track it at same time from various feedbacks. Also the processing unit interfaces with the user whenever required.

C. *Actuators.*

Actuators control the motion of the vehicle and make it traverse the calculated path. The two types of control that is required for this system is steering control and speed control. The steering and the speed actuations are controlled automatically from the processor. This actuations can be

DOI: 01.IJCSI.02.02.130

achieved in a “steer by wire” and “drive by wire” [5-7] embedded cars by simply giving the corresponding signal to the actuators as they are electrically controlled, whereas in mechanical assemblies the control solution will have to be an automaton, where the steer angle and speed are controlled mechanically by actuators (rotary or linear) which are again controlled by the processor [8]. These actuators are a close loop system and provide feedback to the processor to maintain accuracy.

D. User Interface.

The system forms an interactive ambience with the user. Not only it takes inputs like “start parking”, “put reverse gear” and “put parking gear”, the user is the authoritative that can quit the process at any point of time and return back to its normal maneuver. In autonomous parking mode this system displays the trajectory and the localization of the vehicle on the trajectory, elapsed time, localization of interfering things and other such aesthetic parameters on user graphic LCD.

IV. ALGORITHM

As soon as the start command is received from the user interface the parking mission begins. Immediately it decides whether parking is a left side or right side parking and takes this into consideration for future calculations. Next it calculates the angle of misalignment Φ. from the already parked vehicle. Then it locks the steering at perfect straight position and moves a little distance and ensures the angle Φ.

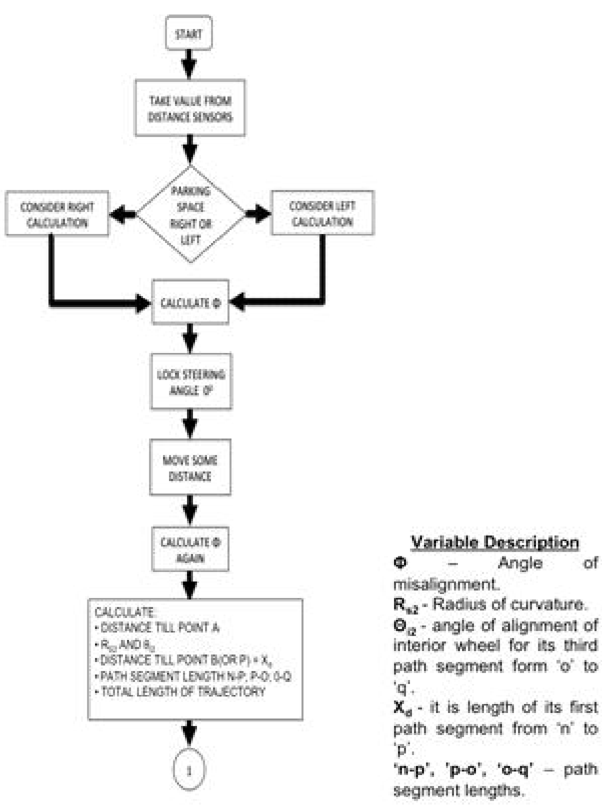

Fig. 8 – flow chart of the algorithm part1

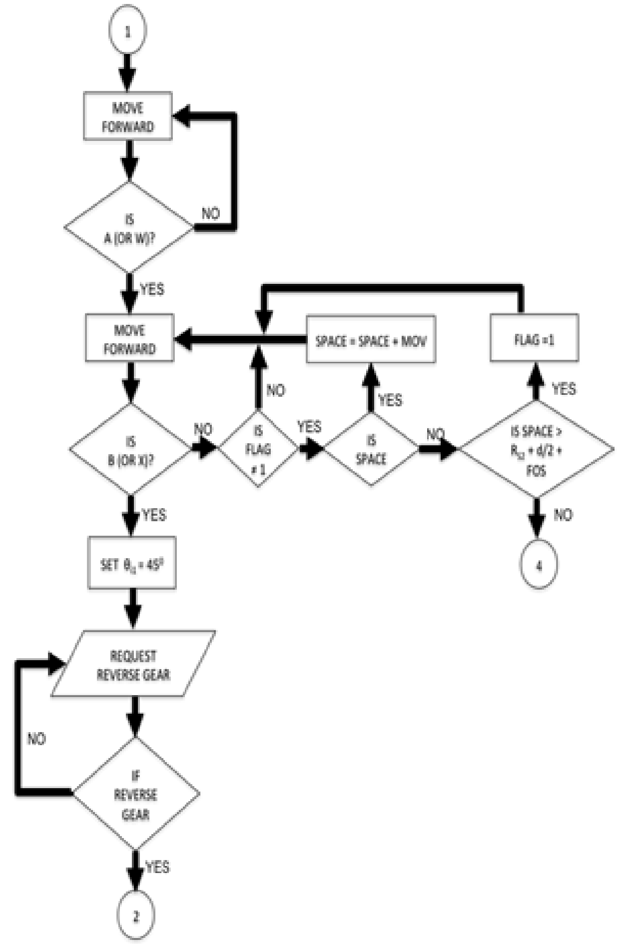

Fig. 9 – flow chart of the algorithm part2

Once the Φ is finalized then all the other calculations are performed taking into consideration the effect of Φ on them. The point A (see fig. 6) is localized and the distance that the vehicle needs to travel to point A is determined. Also the distance till point B (see fig. 6) that is X_d is determined, along with the radius of curvature of second steer-angle R_{s2} and θ_{i2}. Immediately the whole path length from the length of each path segment is determined. After all calculations the boundary conditions are determined, if the conditions fails the user interface displays corrective solution and system exits automatic parking mode or else the vehicle maneuvers on the calculated trajectory, which is tracked by the wheel encoder. Parallel to this, the knowledge of the parking space availability and obstacle on its trajectory is continuously sensed. If the space is not sufficient then it intimates the driver and exits the automatic parking mode and if it detects an obstacle on its path then it goes into delay mode and pauses and continues when the path is cleared.After completing its first segment of trajectory from n to p (see fig. 2), the steering angle is changed such that the angle of inner wheel θ_{i1} is exactly 45^0. Then user interface requests the driver to put the vehicle into reverse gear. After which the front collision sensors sleep and rear collision sensor activates. The vehicle starts traversing on the second segment of its trajectory from point ‘p’ to point ‘o’ (see fig. 20). Then the steering angle is changed again as per the calculated θ_{i2} using (1.a, 1.b) and (2). Then lastly the vehicle moves on the final

DOI: 01.IJCSI.02.02.130

segment of its trajectory o to q and requests to put the vehicle into parking gear to end the mission. During the whole process the driver has the option to end the task and return back to manual mode.

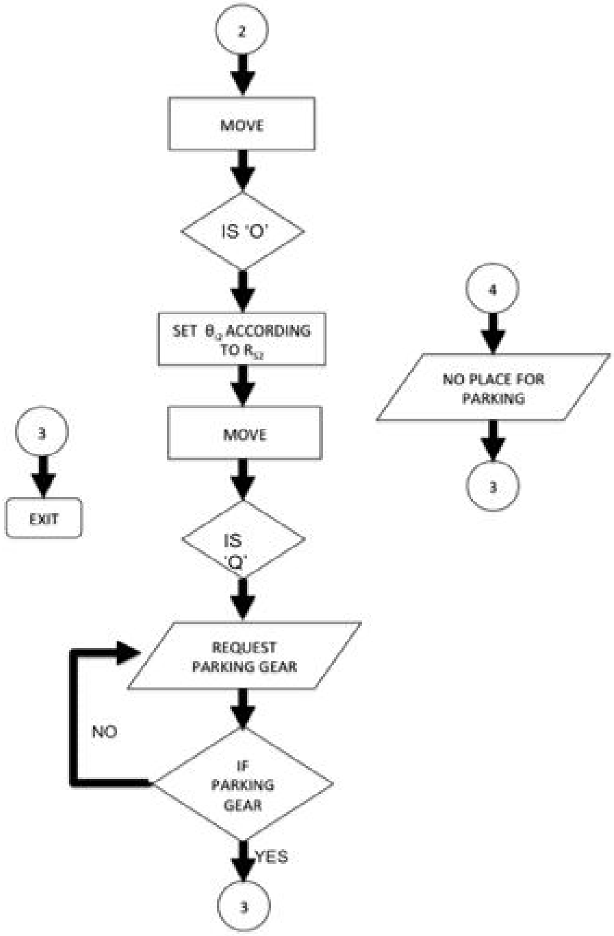

Fig. 10 – Flow chart of the algorithm part 3

inner wheel θ_{i1} is exactly 45^0. Then user interface requests the driver to put the vehicle into reverse gear. After which the front collision sensors sleep and rear collision sensor activates. The vehicle starts traversing on the second segment of its trajectory from point 'p' to point 'o' (see fig. 20). Then the steering angle is changed again as per the calculated θ_{i2} using (1.a, 1.b) and (2). Then lastly the vehicle moves on the final segment of its trajectory o to q and requests to put the vehicle into parking gear to end the mission. During the whole process the driver has the option to end the task and return back to manual mode.

V. SIMULATIONS

The representative of vehicle parameters such as width of the car, length of the car is used for simulation in MATLAB environment. The fig. 11 illustrates the parallel parking trajectory simulation result for the path planning. It is been deduced that the path of the misaligned vehicle (other parameters kept constant) is greater than ideally aligned vehicle and hence driver must attempt to keep his vehicle aligned to the parked car to save the parking time.

Distance

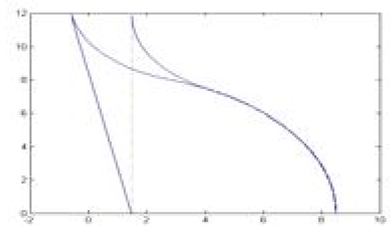

Fig. 11 – Trajectory of the ideal and misaligned vehicle in simulation configuration. (Trajectory dependency on the angle of misalignment)

The fig. 12 illustrates that the trajectory will be shortest if the outer bodies of the vehicles abut each other, which has no practical applicability. Therefore, the vehicle will follow the shortest trajectory when it is as nearer to the parked vehicle. The farthest distance between the vehicles allowed would be the width of the vehicle deducted from the parallel parking space, ideally. Practically factor of safety is considered. Beyond this maximum distance the parking trajectory interferes with the parked vehicle

VI. CONCLUSION AND FUTURE SCOPE

A low cost and staunch autonomous parallel parking method is proposed. The path planning (section 2) derives the vehicles trajectory for a perfectly aligned estimated parking position, irrespective of the orientation and position of the vehicle within its boundary conditions. Simple calculations and limited sensing devices attributes to its ease in processing, low cost and agile performance. This particular algorithm can also be implemented for the perpendicular parking situations with few minor changes.

Distance

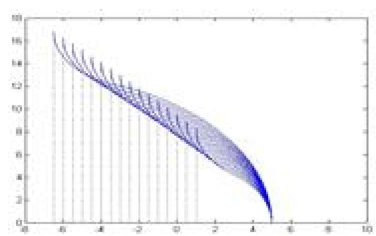

Fig. 12 – Boundary conditions of the path planning in simulation environment. (Trajectory dependency on the distance between the vehicles)

The addition of subsystems would further add to the accuracy of the system. The wheel pressure can be sensed and calibrated to the diameter of the wheel so as increase accuracy of the path tracking. The path tracking can be furthered improved with embedding GPS into the system that will look after the errors due to slip and skid of the wheel. The GPS will also provide a new definition to the calculations of few parameters like Φ and its relatives, which will be much easier and accurate.

ACKNOWLEDGEMENT

Our special thanks to Sanket Deshpande (Maharashtra Institute of Technology, Pune), Rohit Nayak (MIT, Manipal), for their help and concern.

DOI: 01.IJCSI.02.02.130

REFERENCES

[1] Jin Xu, Guang Chen and Ming Xie; " Vision- Guided Automatic Parking for Smart Car," Intelligent Vehicles Symposium, 2000. IV 2000, pp. 735 – 730, Oct 2000.
[2] Allan Bonnick; " Automotive Science and Mathematics"; Butterworth Heinemann, 2008.
[3] Carullo, A. and Parvis, M.; "An ultrasonic sensor for distance measurement in automotive applications"; Sensors Journal, IEEE, vol. 1, issue: 2, pp. 143-147, Aug 2001.
[4] Pascal Desbiolles, Achim Friz; " Development of high resolution sensor element MPS40S and dual track magnetic encoder for rotational speed and position measurement"; NTN Technical Review, NO. 75, 2007.
[5] Joachim Langenwalter; The MathWorks "embedded automotive system development process - steer-by-wire system", Design, Automation and test in Europe, 2005. Proceedings, vol. 1, pp. 538-539, Mar 2005.
[6] S. Laws, C. Gadda, S. Kohn, P. Yih, J.C Gerdes, J.C Milroy, "Steer-by-wire suspension and steering design for controllability and observability", 16th IFAC World Congress, Volume 16, part 1, 2005.
[7] Se - Wook OH, Ho-Chol CHAE, Seok-Chan YUN and Chang-Soo HAN; "The design of a controller for steer-by-wire system"; JSME international Journal, VOl. 47(2004), No. 3, pp. 896 – 907.
[8] Jose E. Naranjo; Carlos Gonzalez; Garcia, R; de Pedro, T.; Haber, R.E.; " power-steering control architecture for automatic driving", IEEE transaction Intelligent Transportation System, VOL.6, NO. 4, pp. 406 – 415, Dec 2005.

DOI: 01.IJCSI.02.02.130